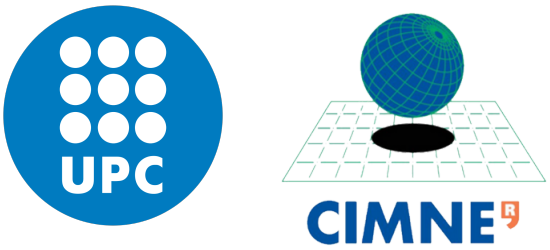
Numerical Integration of Constitutive Damage Models using MATLAB

Computational Solid Mechanics
Assignment - 1

by

Sanath Keshav

Erasmus Mundus Masters of Science in Computational Mechanics



universitat politècnica de catalunya

Barcelona

March 2018

Contents

1 Rate Independent Damage Models - Part I

1.1 Characterization of the Elastic Domain

In addition to the symmetric tension compression model, the tension only model and the non symmetric tension compression model were implemented. All the models are representative in the High-Westergaard stress space. The models vary in their respective definitions of the damage surface.

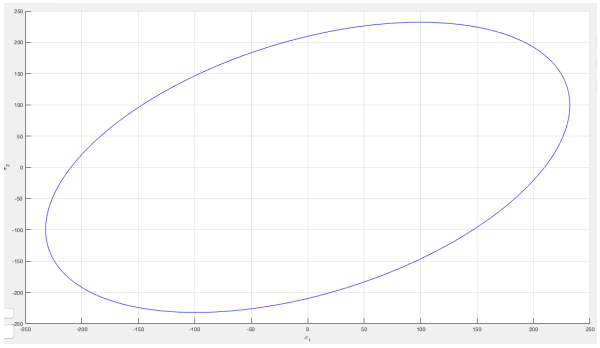


Figure 1a: Symmetric model

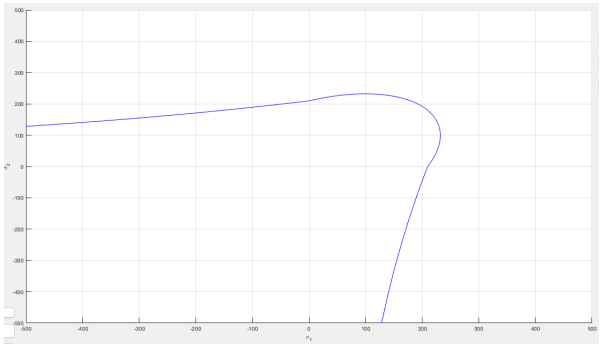


Figure 1b: Tension only model

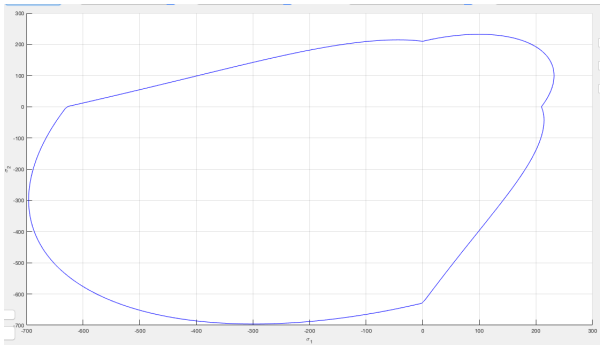


Figure 1c: Non Symmetric Tension Compression model

The elastic region in the first quadrant fo the tension only model and the non symmetric tension compression model are exactly the same as the symmetric case. In the tension only model the second and fourth quadrant behave asymptotically while in the third quadrant is elastic under any biaxial compressive stresses. In the non symmetric tension compression model, it is postulated that the damage happens at much higher stresses in compressive loading compared to tensile loading which can be observed in materials such as concrete etc. This leads to a bigger elastic domain in the third quadrant compared to the first quadrant.

1.2 Characterization of the Hardening / Softening Law

The exponential hardening law was implemented in which there is an exponential response of the hardening variable q with respect to the internal variable r in contrast to the linear hardening law which has a linear response of hardening variable q with respect to the internal variable r.

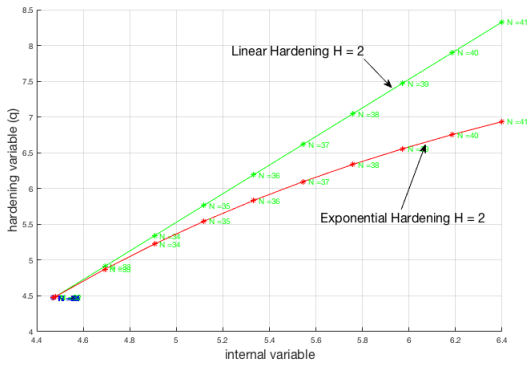


Figure 2a: Linear and Exponential Hardening q vs r

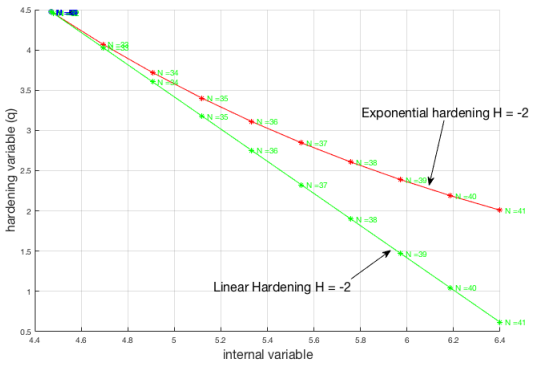


Figure 2b: Linear and Exponential Softening q vs r

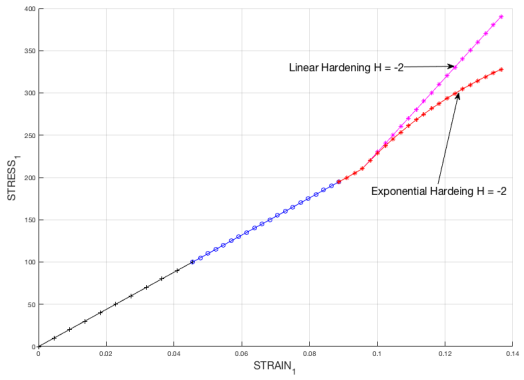


Figure 2c: Linear and Exponential Hardening

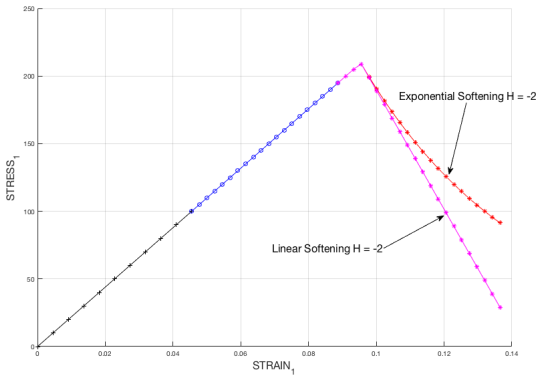


Figure 2d: Linear and Exponential Softening

The exponential and linear hardening / softening is observed by plotting the hardening variable q vs internal variable r for a uniaxial tensile loading case.
It was also observed that the evolution of the damage variable with time was slower in the exponential case compared to the linear case.

1.3 Case 1 - Uniaxial Tension Compression Loading

The correctness of code implementation was rigorously verified by subjecting it into several loading conditions. All cases have been analyzed with the exponential hardening softening law using a particular hardening modulus H as the linear case was already implemented.

The loading path of the first case is given by

$$\begin{aligned} \Delta\bar{\sigma}_1^{(1)} = 250 \quad &: \quad \Delta\bar{\sigma}_2^{(1)} = 0 \\ \Delta\bar{\sigma}_1^{(2)} = -600 \quad &: \quad \Delta\bar{\sigma}_2^{(2)} = 0 \\ \Delta\bar{\sigma}_1^{(3)} = 350 \quad &: \quad \Delta\bar{\sigma}_2^{(3)} = 0 \end{aligned}$$

Exponential Hardening of $H = -1$ and poisson ratio $\nu = 0.3$ was chosen to illustrate all cases in this section.

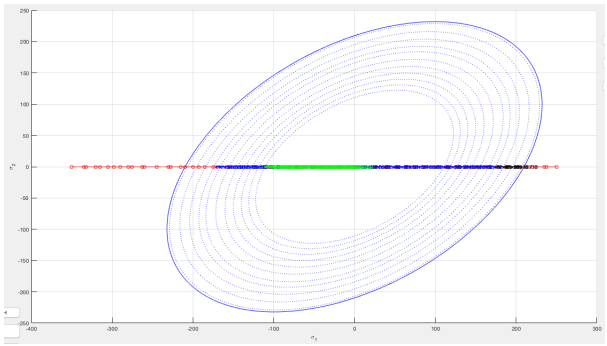


Figure 3a: Symmetric model Loading path and damage surface

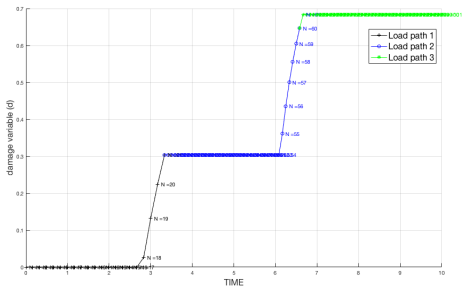


Figure 3b: Damage variable evolution

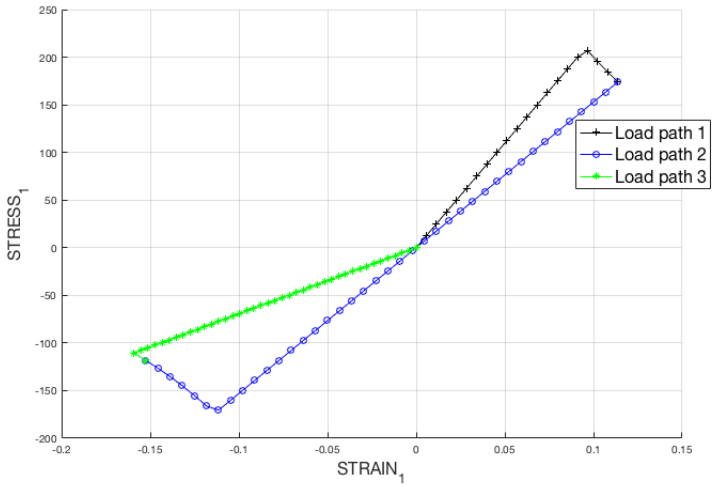


Figure 3c: Stress 1 vs. Strain 1

Appropriate loading paths were chosen to illustrate the effects of tensile and compressive damage. An elastic response is observed in the first loading path until the yield point and damage under tensile loading starts to occur following an exponential softening law. During the second step of tensile unloading and compressive loading an elastic response is observed with a deteriorated "young modulus" followed by damage due to compressive loading which further softens under the exponential softening law. The third step of compressive unloading exhibits a linear elastic response under an even more deteriorated "young modulus" until there is no loading. The evolution of the damage variable with time is also plotted which illustrates the evolution of the damage variable only during the tensile damage in the first step and the compressive damage in the second step

Next, the same loading path and exponential hardening of H = -1 is considered for the tension only model and the non symmetric tension compression model.

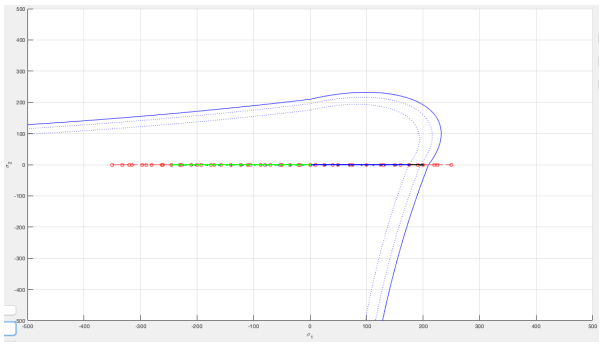


Figure 4a: Tension only model Loading path and damage surface

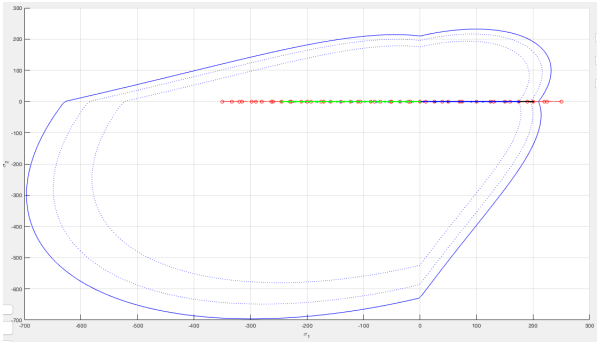


Figure 4b: Non symmetric model damage surface

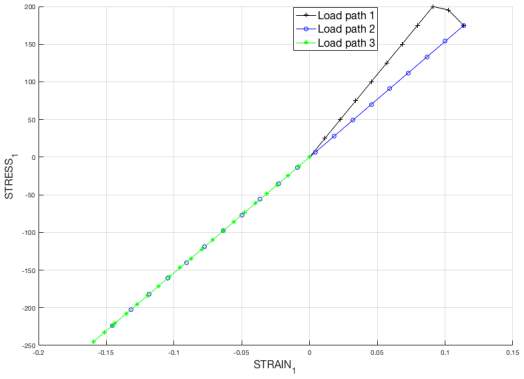


Figure 4c: Stress 1 vs. Strain 1

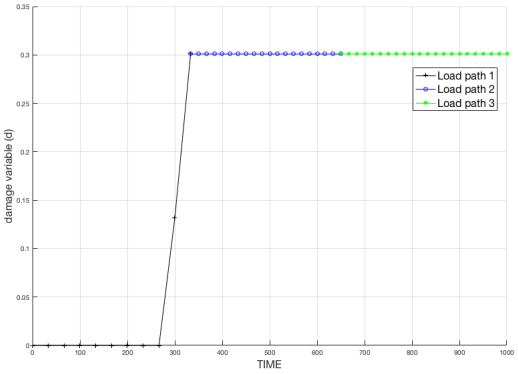


Figure 4d: Damage variable evolution

In accordance to theory, both the Tension only model and the non symmetric tension compression model behave the exact same way under these loading conditions.

In the first load step the same behavior that was seen in the symmetric model of elastic response followed by damage under tensile conditions is observed. In the second tensile unloading and compressive loading load step damage does not occur as we never encounter the damage surface during the compression. Hence an elastic response is observed throughout with the deteriorated "young modulus". The third step of compressive unloading also just provides an elastic response. It is also apparent that the evolution of the damage variable happens during the first step of tensile damage.

1.4 Case 2 - Biaxial Tension Compression Loading

The loading path of the second case of Biaxial tension compression loading is given by

$$\begin{aligned} \Delta\bar{\sigma}_1^{(1)} = 250 \quad &: \quad \Delta\bar{\sigma}_2^{(1)} = 0 \\ \Delta\bar{\sigma}_1^{(2)} = -250 \quad &: \quad \Delta\bar{\sigma}_2^{(2)} = -250 \\ \Delta\bar{\sigma}_1^{(3)} = 100 \quad &: \quad \Delta\bar{\sigma}_2^{(3)} = 100 \end{aligned}$$

Exponential Hardening of $H = -1$ and poisson ratio $\nu = 0.3$ was chosen to illustrate all cases in this section.

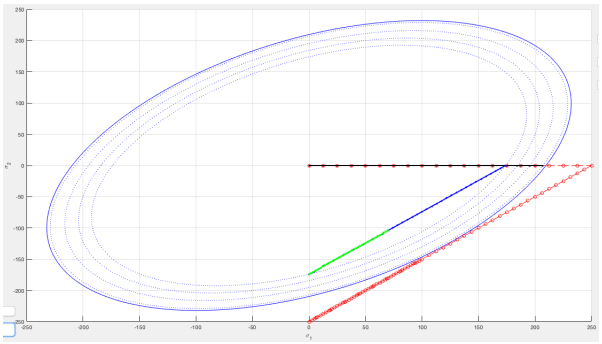


Figure 5a: Symmetric model Loading and damage surface

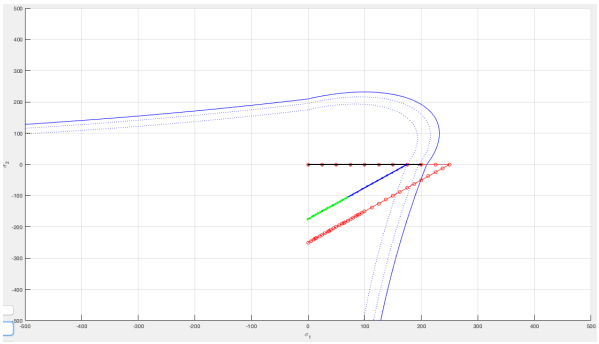


Figure 5b: tension only model Loading and damage surface

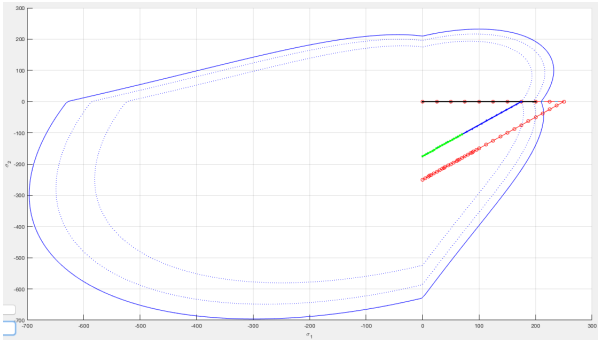


Figure 5c: Non symmetric model Loading path and damage surface

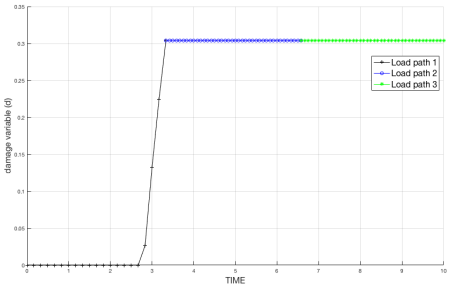


Figure 5d: Damage variable evolution

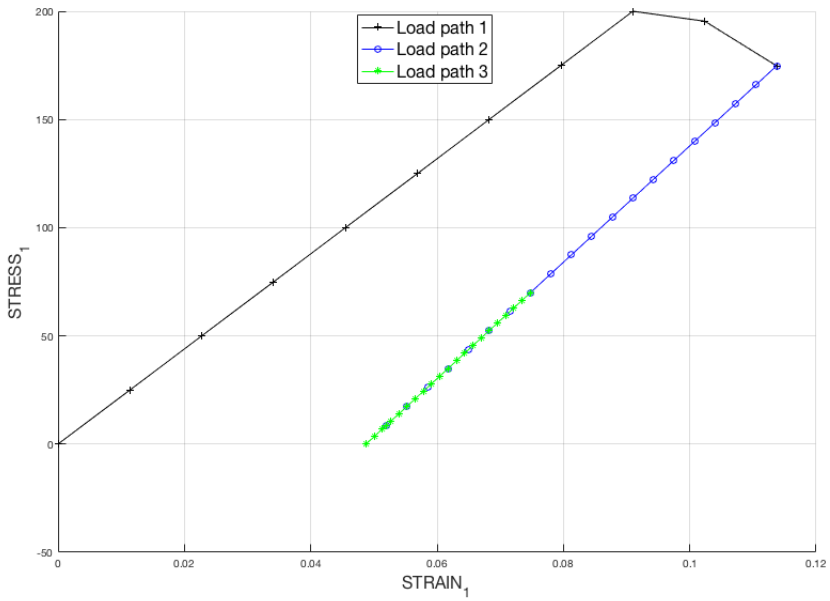


Figure 5e: Stress 1 vs. Strain 1

In accordance to theory, the symmetric model, the Tension only model and the non symmetric tension compression model behave the exact same way under these loading conditions.

The first load step is straightforward uniaxial tensile loading which causes damage and due to the negative hardening modulus causes the contraction of the damage surface in the stress space. Due to this contraction of the damage surface, the second and third load step of biaxial loading elicits an elastic response with a deteriorated constitutive matrix.

It is also interesting to observe the behavior at the end of the second load step when the load in the first principle direction is zero, the strain in the first principle direction is not zero (but small) due to ***poisson*** effects of the loading in the other direction. Hence a plastic like behavior is seen in the stress 1 vs strain 1 graph which has nothing to do with plasticity. Also, the damage happens only in the first load step, we can observe the evolution of the damage variable during the first load step and then remaining constant.

1.5 Case 3 - Biaxial Tension Compression Loading

The loading path of the second case of Biaxial tension compression loading is given by

$$\begin{aligned} \Delta\bar{\sigma}_1^{(1)} = 250 \quad &: \quad \Delta\bar{\sigma}_2^{(1)} = 250 \\ \Delta\bar{\sigma}_1^{(2)} = -600 \quad &: \quad \Delta\bar{\sigma}_2^{(2)} = -600 \\ \Delta\bar{\sigma}_1^{(3)} = 350 \quad &: \quad \Delta\bar{\sigma}_2^{(3)} = 350 \end{aligned}$$

Exponential Hardening of $H = -1$ and poisson ratio $\nu = 0.3$ was chosen to illustrate all cases in this section.

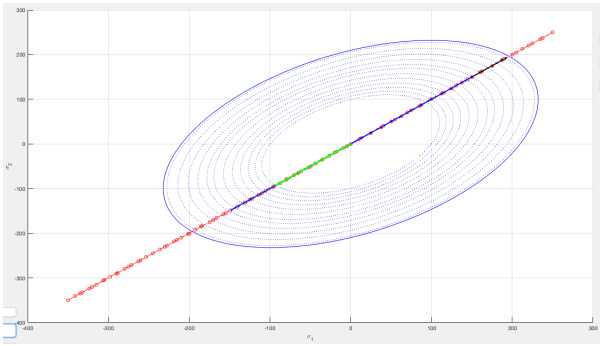


Figure 6a: Symmetric model Loading path and damage surface

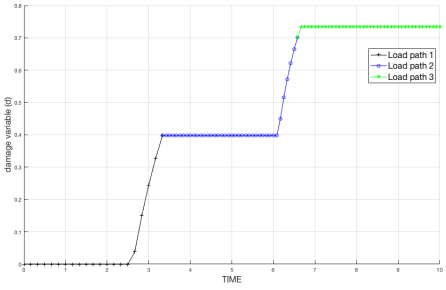


Figure 6b: Damage variable evolution

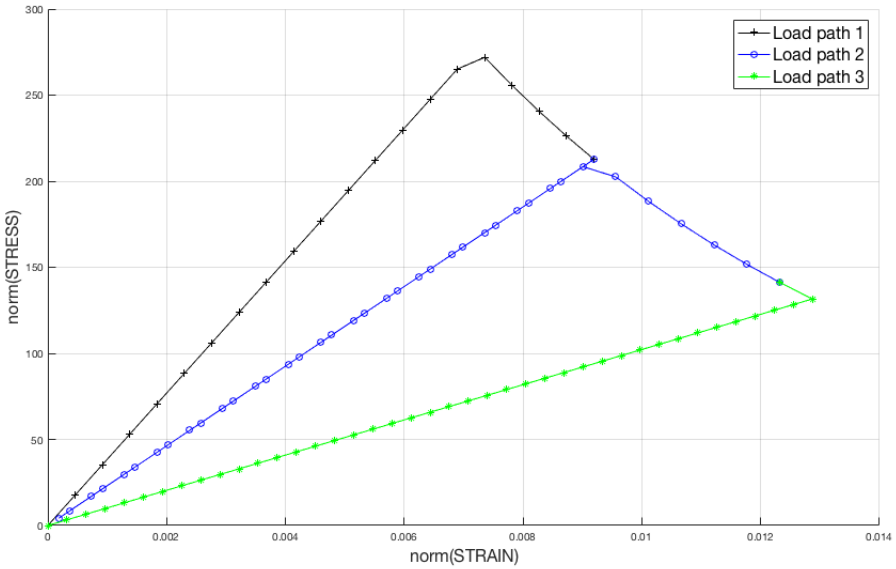


Figure 6c: norm Stress vs. norm Strain

In this loading scenario, a biaxial tensile and compressive load is subjected and the behavior between the norm of the stress and the norm of the strain is plotted. In the first load step, a biaxial tensile load is applied which initially causes elastic behavior following the constitutive tensor until it reaches the damage surface and causes damage under tensile loading conditions. Because if the negative hardening modulus the damage surface contracts until the second load step of biaxial compressive unloading followed by loading begins. During the biaxial unloading, elastic behavior is observed with a deteriorated constitutive tensor because of the damage occurred. It is then followed by compressive biaxial loading which initially shows elastic behavior and then proceeds towards damage under biaxial compressive loading. The third load step shows elastic behavior under biaxial loading with an even further deteriorated constitutive tensor caused due to damage in the two previous stages. The evolution of the damage variable can be observed where damage variable changes first under tensile biaxial loading in the first step and then under compressive biaxial loading.

Next, the same loading path and exponential hardening of H = -1 is considered for the tension only model and the non symmetric tension compression model.

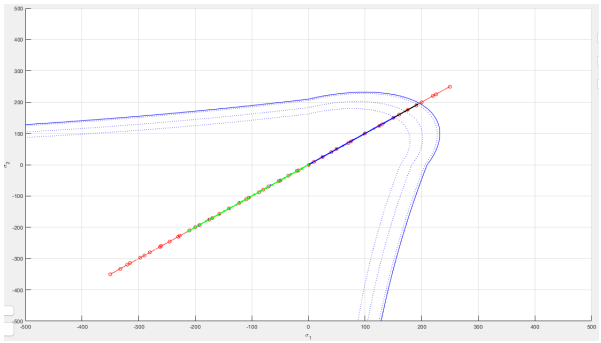


Figure 7a: Tension only model Loading path and damage surface

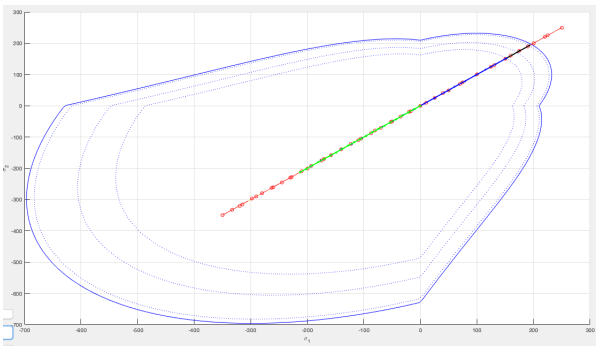


Figure 7b: Non symmetric model damage surface

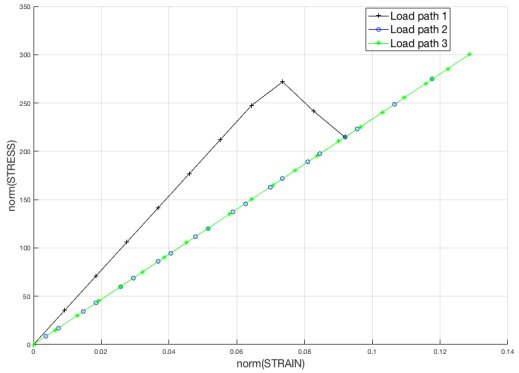


Figure 7c: Stress 1 vs. Strain 1

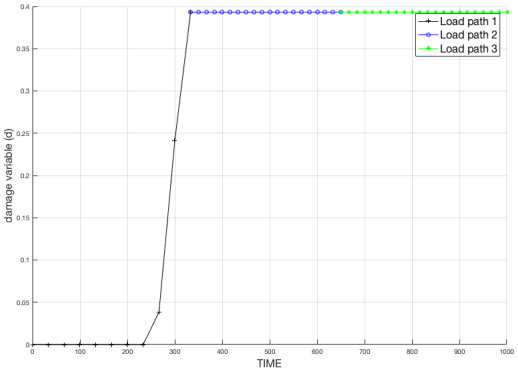


Figure 7d: Damage variable evolution

In accordance to theory, both the Tension only model and the non symmetric tension compression model behave the exact same way under these loading conditions.

The first load step behavior is similar to the symmetric case where an elastic behavior is observed until the damage surface is reached and then damage occurs. Due to the negative hardening modulus the elastic domain shrinks. The second step of biaxial unloading followed by biaxial compressive loading exhibits elastic behavior unlike the symmetric case where damage under compressive biaxial loading was observed as the loading never reaches the damage surface in these cases. Hence they exhibit elastic behavior with a deteriorated constitutive tensor due to damage in the first loading step. The damage variable evolution can also be observed where the damage variable changes only during damage of the first step of tensile biaxial loading.

2 Rate Dependent Damage Models - Part II

2.1 Effects of variation of viscosity parameters

The continuum isotropic visco-damage model was implemented for the plane strain symmetric tension compression model. In this subsection the problem is subjected to uniaxial tension with the following loading paths and parameters.

$$\begin{aligned} \Delta\bar{\sigma}_1^{(1)} = 100 \quad &: \quad \Delta\bar{\sigma}_2^{(1)} = 0 \\ \Delta\bar{\sigma}_1^{(2)} = 100 \quad &: \quad \Delta\bar{\sigma}_2^{(2)} = 0 \\ \Delta\bar{\sigma}_1^{(3)} = 300 \quad &: \quad \Delta\bar{\sigma}_2^{(3)} = 0 \\ H = 0 \quad &, \quad \text{Time Interval} = 1 \\ \nu = 0.3 \quad &, \quad \eta = 0, 0.1, 1, 10 \end{aligned}$$

This choice of hardening modulus $H = 0$ was made to illustrate the phenomenon easily without having to zoom in. The same phenomenon occurs for any H.

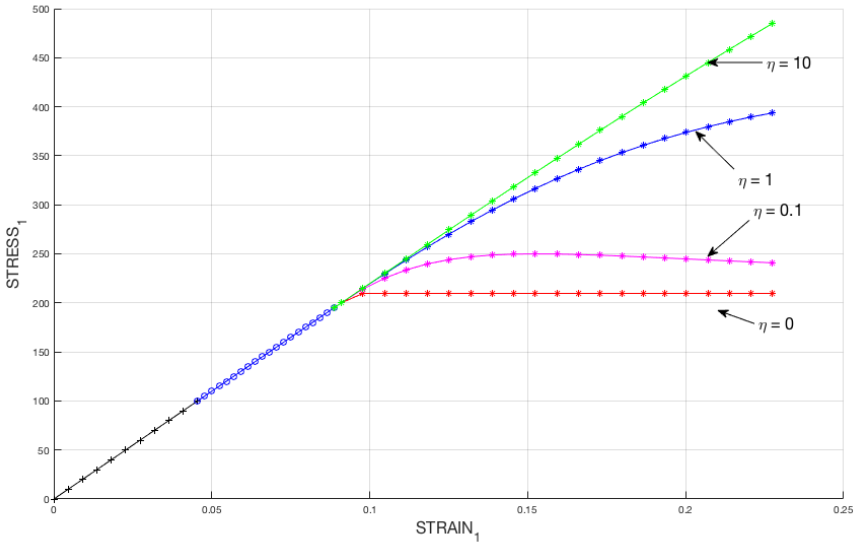


Figure 8: Stress vs. Strain for different viscosities

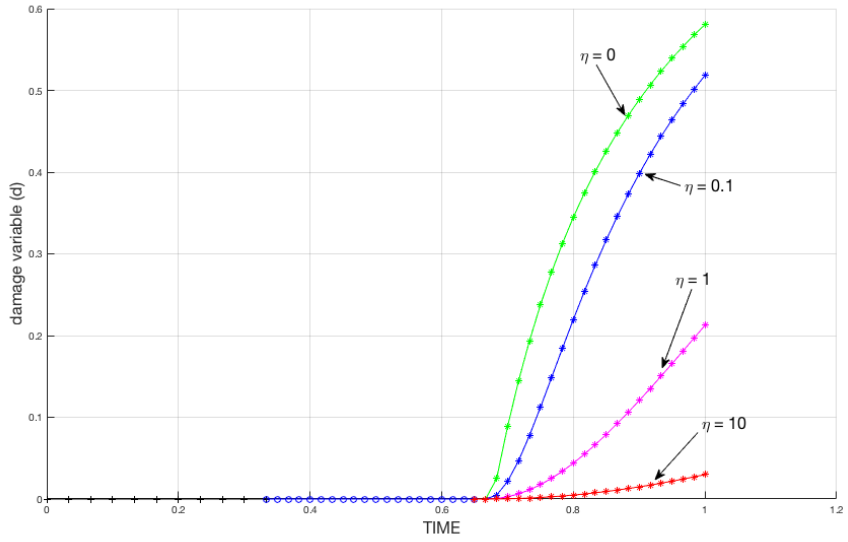


Figure 9: Damage variable evolution for different viscosities

With the variation of the viscosity parameter η the stress strain behavior and the damage variable evolution behavior was plotted. We know that the evolution of the internal variable r is given by an evolution law which is inversely dependent on the viscosity parameter η. Since we assumed that the hardening modulus $H = 0$ and $\dot{q} = H(r)\dot{r}$ and $d = 1 - \frac{q}{r}$, the inverse proportionality of the viscosity parameter η is manifested very clearly as expected with theory in the evolution of the damage variable d. The stress vs strain behavior is also consistent which shows how the most viscous case offers more elastic behavior due to lesser damage as compared to the inviscid case.

2.2 Effects of variation of strain rate

The problem is subjected to uniaxial tensile loading and the strain rates are varied to analyze the behavior of strain rate on the stress strain curve. The strain rate is inversely proportional to the time interval in which the problem is being solved. I this subsection we vary the time interval keeping all the other parameters same. In this subsection the problem is subjected to uniaxial tension with the following loading paths and parameters.

$$\begin{aligned} \Delta\bar{\sigma}_1^{(1)} = 100 \quad &: \quad \Delta\bar{\sigma}_2^{(1)} = 0 \\ \Delta\bar{\sigma}_1^{(2)} = 100 \quad &: \quad \Delta\bar{\sigma}_2^{(2)} = 0 \\ \Delta\bar{\sigma}_1^{(3)} = 300 \quad &: \quad \Delta\bar{\sigma}_2^{(3)} = 0 \\ H = 0 \quad &, \quad \text{Time Interval} = 0.1, 1, 10, 100 \\ \nu = 0.3 \quad &, \quad \eta = 1 \qquad \alpha = 0.5 \end{aligned}$$

This choice of hardening modulus $H = 0$ was made to illustrate the phenomenon easily without having to zoom in. The same phenomenon occurs for any H.

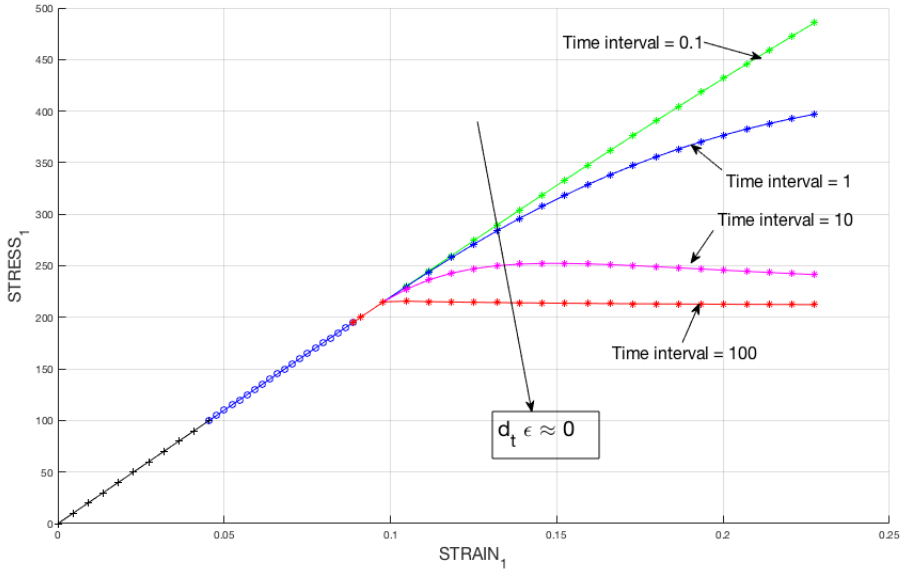


Figure 10: Stress vs. Strain for different strain rates

According to rheological model law $F = \eta\dot{\delta}$, we know that the force is directly proportional to the velocity and the viscosity. Since we have the viscosity parameter constant, we can observe that the stress response is proportional to the strain rate.

2.3 Effects of variation of α

In this subsection we vary the parameter α involved in the integration algorithm and analyze the effects on the stress strain curve and the evolution of the first component of the $\mathbb{C}_{alg}$ and $\mathbb{C}_{tang}$ constitutive tensors keeping all the other parameters same. The problem is subjected to

uniaxial tension with the following loading paths and parameters.

$$\begin{aligned} \Delta\bar{\sigma}_1^{(1)} = 100 \quad &: \quad \Delta\bar{\sigma}_2^{(1)} = 0 \\ \Delta\bar{\sigma}_1^{(2)} = 100 \quad &: \quad \Delta\bar{\sigma}_2^{(2)} = 0 \\ \Delta\bar{\sigma}_1^{(3)} = 300 \quad &: \quad \Delta\bar{\sigma}_2^{(3)} = 0 \\ \text{Exponential Hardening -}H = -0.1 \quad &, \quad \text{Time Interval} = 100 \\ \nu = 0.3 \quad &, \quad \eta = 1 \quad \alpha = 0, 0.25, 0.5, 0.75, 1 \end{aligned}$$

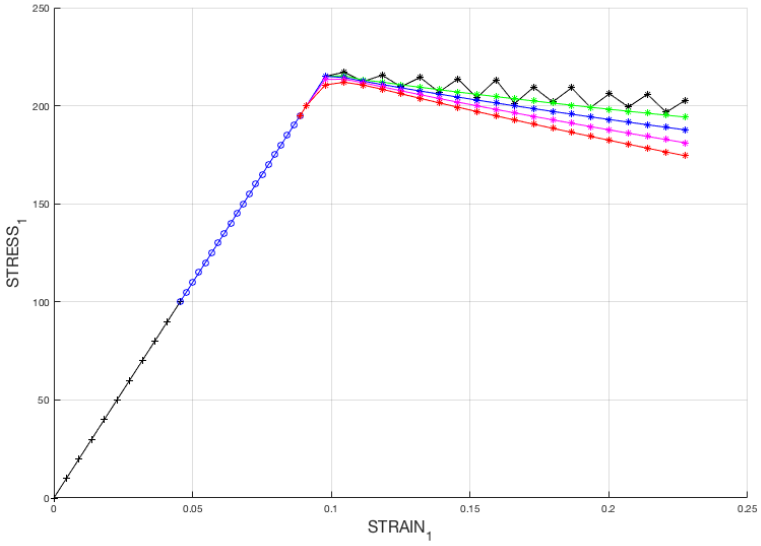


Figure 11: Stress vs. Strain for different α

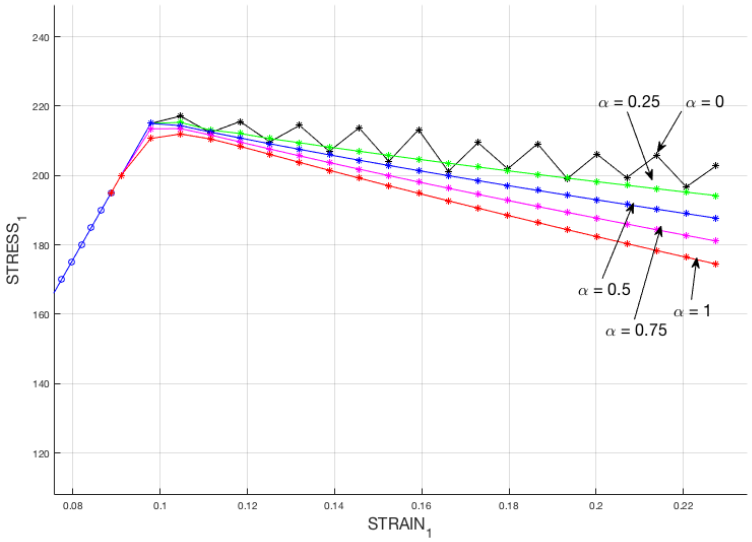


Figure 12: Stress vs. Strain for different α

A large time interval was taken so that the delta t is higher to show the effects of the parameter alpha on the integration algorithm. It can be observed that for $\alpha < 0.5$ oscillatory solutions are produced in accordance to the theory which guarantees unconditional stability for $\alpha \in [0.5, 1]$ and only conditional stability for $\alpha \in [0, 0.5)$. We also know that $\alpha = 0$ corresponds to the forward euler method and $\alpha = 1$ corresponds to the backward euler method which both first order accurate methods. $\alpha = 0.5$ corresponds to the Crank Nicolson method which is a second order accurate method.

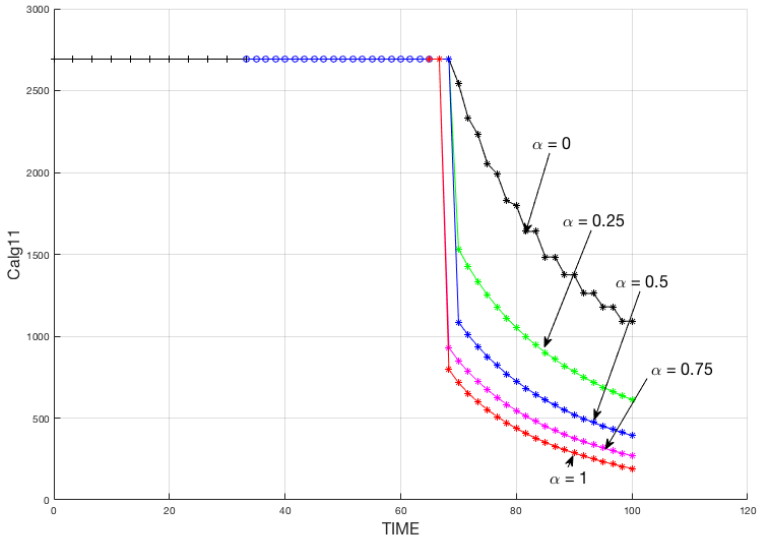


Figure 13: $\mathbb{C}_{11}^{alg}$ vs Time for different α

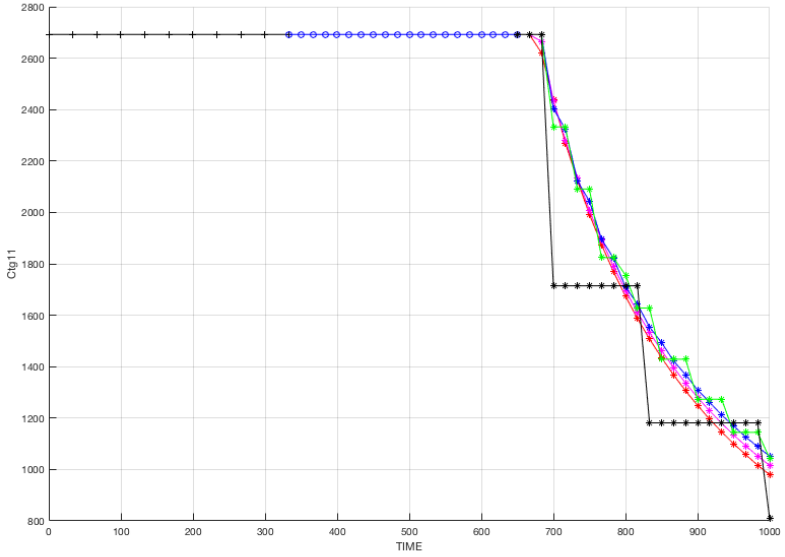


Figure 14: $\mathbb{C}_{11}^{tang}$ vs Time for different α

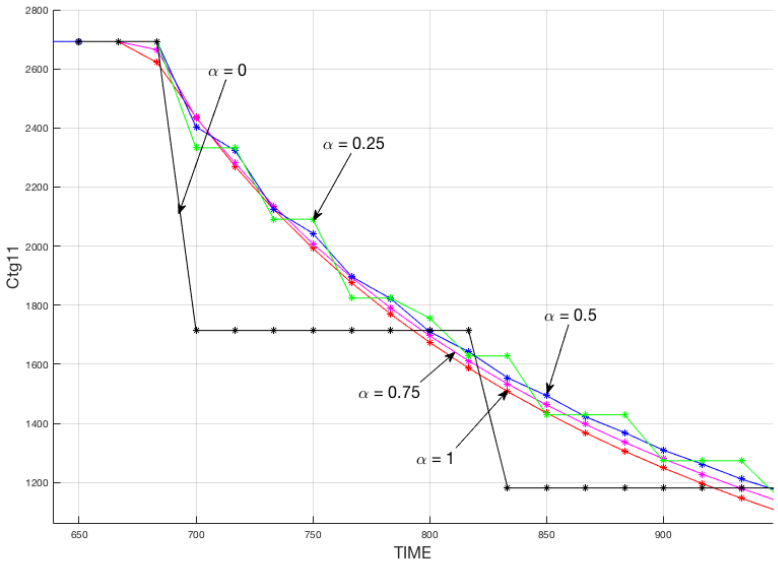


Figure 15: $\mathbb{C}_{11}^{tang}$ vs Time for different α

With variation of α, similar behavior of unstable oscillatory solutions were observed for $\alpha = 0, 0.25$ in the evolution of the first component of the Algorithmic constitutive tensor as well as the tangent constitutive tensor. An expected discontinuity is observed in the $\mathbb{C}_{11}^{alg}$ vs Time graph. It was also observed that for $\alpha = 0$ the $\mathbb{C}_{11}^{alg}$ and $\mathbb{C}_{11}^{tang}$ were same as expected. It was also observed that for $\alpha = 1$ and $\eta = 0$ the inviscid case was recovered.

3 APPENDIX

3.1 dibujar_criterio_dano1.m

```
function hplot = dibujar_criterio_dano1(ce,nu,q,tipo_linea,MDtype,n)
ce_inv=inv(ce);
c11=ce_inv(1,1);
c22=ce_inv(2,2);
c12=ce_inv(1,2);
c21=c12;
c14=ce_inv(1,4);
c24=ce_inv(2,4);

% POLAR COORDINATES
if MDtype==1
    tetha=[0:0.01:2*pi];
    D=size(tetha);
    m1=cos(tetha);
    m2=sin(tetha);
    Contador=D(1,2);


    radio = zeros(1,Contador) ;
    s1    = zeros(1,Contador) ;
    s2    = zeros(1,Contador) ;

    for i=1:Contador
        radio(i)= q/sqrt([m1(i) m2(i) 0 nu*(m1(i)+m2(i))]*ce_inv*[m1(
            i) m2(i) 0 ...
            nu*(m1(i)+m2(i))]');

        s1(i)=radio(i)*m1(i);
        s2(i)=radio(i)*m2(i);

    end
    hplot =plot(s1,s2,tipo_linea);


elseif MDtype==2

    tetha=[-pi/2 + 0.01: 0.01 : pi - 0.01];
    m1=cos(tetha);
    m2=sin(tetha);
    m3=m1+m2;
    Contador=size(tetha,2);

    radio = zeros(1,Contador) ;
    s1    = zeros(1,Contador) ;
    s2    = zeros(1,Contador) ;
```

```
45      m1plus = m1(:).*(m1(:)>0);
46      m2plus = m2(:).*(m2(:)>0);
47      m3plus = m3(:).*(m3(:)>0);
48      for i = 1:Contador
49          radio(i)=q/sqrt([m1plus(i) m2plus(i) 0 nu*m3plus(i)]*ce_inv*[
                m1(i) m2(i) 0 nu*m3(i)]');
50          s1(i)=radio(i)*m1(i);
51          s2(i)=radio(i)*m2(i);
52      end
53      hplot =plot(s1,s2,tipo_linea);
54
55
56  elseif MDtype==3
57      tetha=[0:0.01:2*pi];
58      m1=cos(tetha);
59      m2=sin(tetha);
60      m3=m1+m2;
61      Contador=size(tetha,2);
62
63      radio = zeros(1,Contador) ;
64      s1    = zeros(1,Contador) ;
65      s2    = zeros(1,Contador) ;
66      m1plus = m1(:).*(m1(:)>0);
67      m2plus = m2(:).*(m2(:)>0);
68      for i = 1:Contador
69          ratio = sum([m1plus(i) m2plus(i)])/sum(abs([m1(i) m2(i)]));
70          radio(i)=q/((ratio + (1-ratio)/n)*sqrt([m1(i) m2(i) 0 nu*m3(i
                )]*ce_inv*[m1(i) m2(i) 0 nu*m3(i)]'));
71          s1(i)=radio(i)*m1(i);
72          s2(i)=radio(i)*m2(i);
73      end
74      hplot =plot(s1,s2,tipo_linea);
75  end
76
77  return
```

3.2 Modelos_de_dano1.m

```
1   function [rtrial] = Modelos_de_dano1 (MDtype,ce,eps_n1,n)
2
3   if (MDtype==1)       %* Symmetric
4       rtrial= sqrt(eps_n1*ce*eps_n1');
5
6   elseif (MDtype==2)  %* Tension only
7       sigma_b = ce*eps_n1';
8       sigma_b_plus(:) = sigma_b(:).*(sigma_b(:)>0);
9       rtrial= sqrt(sigma_b_plus*eps_n1');
10
11  elseif (MDtype==3)  %* Non symmetric tension compression
12      sigma_b = ce*eps_n1';
```

```
13      sigma_b_plus(:) = sigma_b(:).*(sigma_b(:)>0);
14      ratio = (sigma_b_plus(1)+sigma_b_plus(2))/(abs(sigma_b(1))+abs(
            sigma_b(2)));
15      rtrial= (ratio+(1-ratio)/n)*sqrt(eps_n1*ce*eps_n1');
16
17  end
18  return
```

3.3 rmap_dano1.m

```
1   function [sigma_n1,hvar_n1,aux_var] = rmap_dano1 (eps_n1,eps_n,hvar_n
        ,Eprop,ce,MDtype,n,viscpr,delta_t)
2
3   hvar_n1 = hvar_n;
4   r_n     = hvar_n(5);
5   q_n     = hvar_n(6);
6   E       = Eprop(1);
7   nu      = Eprop(2);
8   H       = Eprop(3);
9   sigma_u = Eprop(4);
10  hard_type = Eprop(5) ;
11  eta = Eprop(7);
12  ALPHA = Eprop(8);
13
14  %*        initializing
15  r0 = sigma_u/sqrt(E);
16  zero_q=1.d-6*r0;
17
18  %*        Damage surface
19  [rtrial] = Modelos_de_dano1 (MDtype,ce,eps_n1,n);
20  [rtrial_n] = Modelos_de_dano1 (MDtype,ce,eps_n,n);
21  rtrial_nalpha = (1-ALPHA)*rtrial_n + ALPHA*rtrial;
22
23  fload=0;
24
25  if viscpr == 1
26      if(rtrial_nalpha > r_n)
27          %*    Loading
28
29          fload=1;
30          delta_r=rtrial_nalpha-r_n;
31          r_n1= ((eta - delta_t*(1-ALPHA))/(eta + ALPHA*delta_t))*r_n +
                ...
32              ((delta_t)/(eta + ALPHA*delta_t))*rtrial_nalpha;
33          if hard_type == 0
34              %  Linear
35              q_n1= q_n+ H*delta_r;
36              H_n1 = H;
37          else
38              %Exponential
```

```
39              q_inf=r0+(r0-zero_q);
40              if H>0
41                  q_n1=q_n+((H*(q_inf-r0)/r0)*exp(H*(1-rtrial/r0)))*
                        delta_r;
42                  H_n1 = ((H*(q_inf-r0)/r0)*exp(H*(1-rtrial/r0)));
43              else
44                  q_n1=q_n+((H*(q_inf-r0)/r0)*(1/exp(H*(1-rtrial/r0))))
                        *delta_r;
45                  H_n1 = ((H*(q_inf-r0)/r0)*(1/exp(H*(1-rtrial/r0))));
46
47              end
48          end
49
50          if(q_n1<zero_q)
51              q_n1=zero_q;
52          end
53
54
55      else
56
57          %*      Elastic load/unload
58          fload=0;
59          r_n1= r_n   ;
60          q_n1= q_n   ;
61
62
63      end
64
65  else
66      if(rtrial > r_n)
67          %*    Loading
68
69          fload=1;
70          delta_r=rtrial-r_n;
71          r_n1= rtrial   ;
72          if hard_type == 0
73              %  Linear
74              q_n1= q_n+ H*delta_r;
75              H_n1 = H;
76          else
77              %Exponential
78              q_inf=r0+(r0-zero_q);
79              if H>0
80                  q_n1=q_n+((H*(q_inf-r0)/r0)*exp(H*(1-rtrial/r0)))*
                        delta_r;
81                  H_n1 = ((H*(q_inf-r0)/r0)*exp(H*(1-rtrial/r0)));
82              else
83                  q_n1=q_n+((H*(q_inf-r0)/r0)*(1/exp(H*(1-rtrial/r0))))
                        *delta_r;
```

```
84                  H_n1 = ((H*(q_inf-r0)/r0)*(1/exp(H*(1-rtrial/r0))));
85              end
86          end
87
88          if(q_n1<zero_q)
89              q_n1=zero_q;
90          end
91
92
93      else
94
95          %*      Elastic load/unload
96          fload=0;
97          r_n1= r_n   ;
98          q_n1= q_n   ;
99
100
101     end
102 end
103 % Damage variable
104 % ----------------
105 dano_n1    = 1.d0-(q_n1/r_n1);
106 %  Computing stress
107 %  ***************
108 sigma_n1   =(1.d0-dano_n1)*ce*eps_n1';
109
110 % calculation of the Ce_tang_n1
111 if viscpr == 1
112     if rtrial_nalpha > r_n
113         %Algorithm Constitutive Tangent Matrix
114         Ce_alg_n1 = (1.d0-dano_n1)*ce+((ALPHA*delta_t)/(eta+ALPHA*
                delta_t))*...
115             (1/rtrial_nalpha)*((H_n1*r_n1-q_n1)/(r_n1^2))*((ce*eps_n1
                ')'*(ce*eps_n1'));
116         C_alg = Ce_alg_n1(1,1);
117         %Constitutive Tangent Matrix
118         Ce_tan_n1=(1.d0-dano_n1)*ce;
119         C_tan = Ce_tan_n1(1,1);
120
121     else
122         %Algorithm Constitutive Tangent Matrix
123         Ce_alg_n1 = (1.d0-dano_n1)*ce;
124         C_alg = Ce_alg_n1(1,1);
125         %Constitutive Tangent Matrix
126         Ce_tan_n1 = Ce_alg_n1;
127         C_tan = C_alg;
128     end
129 else
130     if rtrial > r_n
```

```
131             Ce_tan_n1= (1.d0-dano_n1)*ce+ ...
132                 (1/rtrial)*((H_n1*r_n1-q_n1)/(r_n1^2))*((ce*eps_n1')'*(ce
                        *eps_n1'));
133             C_tan = Ce_tan_n1(1,1);
134         else
135             Ce_tan_n1 = (1.d0-dano_n1)*ce;
136             C_tan = Ce_tan_n1(1,1);
137         end
138     end
139
140     hvar_n1(5)= r_n1 ;
141     hvar_n1(6)= q_n1 ;
142     hvar_n1(7)= dano_n1 ;
143     if viscpr == 1
144         hvar_n1(8)= C_alg ;
145         hvar_n1(9)= C_tan ;
146     end
147     %* Auxiliar variables
148     aux_var(1) = fload;
149     aux_var(2) = q_n1/r_n1;
```